18.433 final. This exam is closed book. You can have one double-sided handwritten sheet of paper with anything you want on it. Be neat! **In any problem, you can refer to results we have covered in class, but you need to state them precisely.** If you need more space for a question, you can continue on one of the extra pages at the end, but write a pointer to it.

Your Name:

1. Consider a directed graph $G = (V, E)$ with nonnegative (upper) capacities $u : E \to \mathbb{R}$ (and no lower capacities). For any two vertices $s, t \in V$, define $\lambda_{st} \in \mathbb{R}$ to be the maximum flow value from s to t. Given any 3 vertices $s, t, u \in V$, show that $\lambda_{su} \geq \min(\lambda_{st}, \lambda_{tu})$.

2. Suppose we are using the ellipsoid algorithm to find a point in a polytope $P \subseteq \mathbb{R}^n$. Suppose we are given a ball B_0 of radius R containing P and we are told that P is non-empty and contains a ball of radius R/k. Give an upper bound (as a function of n and k) on the number of iterations the ellipsoid algorithm will take to find a point in P. (State precisely any results you use.)

3. Describe a tour-improvement heuristic and a tour-construction heuristic for the traveling salesman problem.

4. Let $M = (E, \mathcal{I})$ be a matroid. Suppose we have a map f from E to $S = \{1, 2, \cdots, k\}$, and define $\mathcal{J} = \{f(I) | I \in \mathcal{I}\}$ (where $f(I) = \{f(e) | e \in I\} \subseteq S$). Show that $(S, \mathcal{J})$ also defines a matroid. (Notationwise this means that we have a partition of E into $E_1, \cdots, E_k$ where $E_j = f^{-1}(j) = \{e \in E : f(e) = j\}$ and with $\mathcal{J} = \{J \subseteq S | \exists I \in \mathcal{I}$ with $I \cap E_j \neq \emptyset$ for all $j \in J\}$.)

5. Let $M = (E, \mathcal{I})$ be a matroid with rank function r and suppose we have a cost function $c : E \rightarrow \mathbb{R}_{\geq 0}$ (for simplicity we are assuming that all the costs are positive). We are interested in finding a base B of maximum total cost, i.e. maximizing $\sum_{e \in B} c(e)$, and we hope to derive algorithms different from the one seen in lecture. **Solve one of the following subquestions (a) or (b).**

   (a) Consider the following greedy algorithm, different from the one covered in lecture.

   - ▷ Sort the elements (from smallest to largest) such that $c(e_1) \leq c(e_2) \leq \cdots \leq c(e_m)$ where $m = |E|$
   - ▷ $S = E$
   - ▷ For $j = 1$ to m
     - ▷ if $r(S \setminus \{e_j\}) = r(E)$ then $S \leftarrow S \setminus \{e_j\}$
   - ▷ Output S

   Does this algorithm return a maximum cost basis in the matroid? Prove it, or give a counterexample.

   (b) Consider the following local search algorithm. Start from any base of M. At any point, define the neighborhood $N(B)$ of a base B to be those bases that can be obtained from B by adding an element in $E \setminus B$ and removing an element of B (so as to maintain a base). Keep replacing the base with a maximum weight base in its neighborhood (and stop whenever the current base is of maximum weight in its neighborhood).

   Is this an *exact* neighborhood, in the sense that whenever this local search algorithm terminates, we are guaranteed to have a maximum base? Explain. State precisely any result you use from the class. (There are several ways to approach this; one approach may involve the exchange graph.)

6. A *derangement* on $\{1, \cdots, n\}$ is a permutation σ such that $\sigma(i) \neq i$ for all $i \in \{1, \cdots, n\}$. There are no derangements for $n = 1$, only one derangement for $n = 2$ (namely $\sigma = (2, 1)$) and only two derangements for $n = 3$ ($\sigma = (2, 3, 1)$ or $\sigma = (3, 1, 2)$). Suppose we are given costs $c(i)$ for $i \in \{1, \cdots, n\}$, and our goal is to find a derangement σ on $\{1, \cdots, n\}$ minimizing $\sum_{i=1}^{n} c(i)\sigma(i)$. Give a polynomial-time algorithm for this problem (there is no need to give the most efficient algorithm, but the algorithm should be polynomial). (You can refer to any algorithm we have seen in class.)